ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 1834–1843

Procedia
Computer Science

International Conference on Computational Science, ICCS 2011

Computational method for agent-based E-commerce negotiations with adaptive negotiation behaviors

Gong Wang[a, *], T. N. Wong[a], Chunxia Yu[a]

[a]*Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong*

Abstract

This paper presents a computational method to organize agent-based E-commerce negotiations with adaptive negotiation behaviors aiming at enhancing the negotiation power and flexibility of software agents to alleviate human involvements in E-commerce negotiations. Firstly, the computational expression of E-commerce negotiation, including negotiation issues and strategies, is specified to assist agents' computing functions. Then, an adaptive negotiation behavior configuration mechanism is proposed to tackle the negotiation dynamics through computation. In this three-staged mechanism, agents' negotiation behaviors are deployed by a case-based strategy assignment mechanism before the starting of negotiation; then along the on-going negotiation sequence, opponents' negotiation behaviors are tracked through Back-Propagation Neural Network (BP_NN) learning model to make strategy adjustment to confront the opponent. After the negotiation, opponents' concession functions are recorded and analysed using time series measure. Finally, the feasibility of the BP_NN learning model is verified through a set of tests. The computational negotiation method is exemplified using a two-issue buyer-seller negotiation case. The outcomes show that the adaptive negotiation behavior configuration mechanism can benefit an agent to win more in the E-commerce negotiation.

Keywords: E-commerce; negotiation; agent; case-based reasoning; neural network

1. Introduction

Negotiation is an effective communication approach to solve transaction conflicts and make better deals between trading entities in the commerce world. Under the flourishing development of E-commerce, negotiations should also be moved to the electronic channel. E-commerce negotiations involve different forms of B2B (e.g., bidding for contracts in a virtual enterprise), B2C (e.g., deliberating customized options in an online shop operated by a company) and C2C (e.g., bargaining price with an individual in a C2C website platform). Human operation-based E-commerce negotiations always relate to iterative online interactions with delayed waiting time. If this function can be designated to some kind of software entity, it will be a preferable situation to alleviate human efforts and reduce the time consumed. Concerning this potential, agent technology has been increasingly explored to automate negotiations [1-3]. A software agent is a computational entity which perceives, acts upon its environment, and is

* Corresponding author. Tel.: +852-6909-8236.
E-mail address: wanggong@hku.hk.

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.199

autonomous in its behavior [4]. Equipped with negotiation functionalities, agents can perform on behalves of their owners to make decisions. Some agent-assisted E-commerce negotiation test beds, such as the MIT Media Lab's Kasbah [5], the Michigan InternetAuctionBot [6] and the MAGNET [7] have provided the vision of using software agents to conduct one-round auction-type negotiations. These early attempts focus more on the illustration of the whole market mechanism with allocation of agent roles, while the behavior dynamics of agents are rather simple. For further development and achieving better performance in multi-round iterative negotiations, agents' negotiation behaviors should be adaptive and dynamic towards various and changeable negotiation environments.

As computational entities, agents' negotiation behaviors are based on some computational models. There are three fundamental areas need to be considered when designing a computational negotiation model: negotiation issues, negotiation protocol and negotiation strategies [1]. Negotiation issues confine the negotiation items on which mutual agreement is aimed to be achieved. The negotiation protocol defines the "rule of encounter" between agents [8]. Negotiation strategies guide agents' negotiation behaviors through combinations of tactics [1]. Since multiple negotiation parties (e.g., one-buyer-one-seller, one-buyer-many-sellers or many-buyers-one-seller) and many negotiation issues (e.g., price, lead time, warranty, etc.) are always involved in E-commerce negotiations, some bilateral or multi-lateral agent negotiation models embracing certain negotiation protocols have been proposed to govern multi-issue negotiations [8-10]. Among these negotiation models, the concurrent one-to-one iterative bidding negotiation model is an effective method to organize multiple agents taking the roles of buyers or sellers. However, concerning an individual negotiation agent, its negotiation behavior is simple and non-adaptive in these models, which means its negotiation strategy is not decided according to specific negotiation environment and cannot be readjusted towards on-going negotiation dynamics. Since most E-commerce negotiations are not fully cooperative and are of win-lose relationships, adaptive negotiation behavior will benefit a negotiation agent to win more against its opponents through more delicate observation of the negotiation situation.

Current researches have developed some learning models facilitating agents' adaptive negotiation behaviors. These learning models are mostly based on the learning and prediction of the negotiation opponent's behaviors, using methods such as the neural network, nonlinear regression analysis and hierarchical clustering [11-14]. However, the offline-learning and online-learning models are separately treated; thereby the accuracy of the learning models and the real-time response to dynamic changes cannot be achieved at the same time. Meanwhile, further processing of the learning data have not been examined to nurture new cycles of negotiations.

The objective of this paper is to configure a generic computational model for agent-based automated negotiations to alleviate human involvements in E-commerce negotiations. The adaptability of negotiation behaviors in the model is enhanced to benefit the negotiation agent to win more against its opponents through more delicate observation of the negotiation situation. The model can be used to build negotiation functionalities of an individual negotiation agent involved in any form of E-commerce negotiations (B2B, B2C and C2C). In this model, both quantitative and qualitative negotiation issues are expressed in a computable manner. A three-staged mechanism is proposed to configure agents' negotiation behaviors adaptively throughout an entire negotiation cycle. In the pre-negotiation stage, agents' negotiation behaviors are deployed by a case-based strategy assignment mechanism. In the on-going negotiation stage, negotiation opponents' behaviors are tracked through the neural network and the self negotiation strategy may be adjusted according to the tracking results. In the post-negotiation stage, negotiation opponents' concession functions will be recorded and analysed using time series measure for future negotiation references. The rest of the paper is organized as follows. Section 2 introduces the computational expression of negotiation issues and strategies in agent-based E-commerce negotiations. Section 3 illustrates the three-staged adaptive negotiation behavior configuration mechanism. Experiments and tests are presented in section 4. Finally, conclusions are drawn in section 5.

2. Computational Expression for E-commerce Negotiations

To enable agents to negotiate automatically through computing functions, all the related negotiation knowledge such as negotiation issues, negotiation strategies and concession tactics must be expressed in a computable manner.

2.1. Negotiation issues

In E-commerce negotiations, buyers and sellers often need to negotiate over a set of considering terms to attain a mutual acceptable solution. Price is the most frequently concerned issue. Sellers always announce a higher price, while buyers desire lower price. Lead time is another important issue when goods are not immediately available and need some time to be delivered. Buyers may need the goods to be delivered as soon as possible, while sellers may require longer time to prepare them. When the product quality issue is considered, at a certain price level, buyers require products with higher quality or more advanced functions, while sellers can only afford products with lower quality or limited functions. Other negotiation issues may involve the payment pattern, the transportation method, the package mode, the penalty terms and so on. To make a deal or contract, agreements must be achieved upon all the negotiation issues. Multiple issues can be negotiated concurrently by packaging one set of issues into a proposal (sent from the seller to buyer) or counter-proposal (sent from the buyer to seller). A negotiation sequence involving multiple concession rounds is composed of alternate placements of proposals and counter-proposals which have to be evaluated to decide their acceptability. Table 1 displays the expressions of related negotiation elements.

Table 1. Negotiation element expressions

Negotiation element	Expression	Example
Issue	I^i $\quad$ $i \in \{\text{negotiation issues}\}$	$I^{price}, I^{leadtime}, I^{quality}$ $\quad$ $i \in \{\text{price, leadtime, quality}\}$
Issue value	Seller: $I(s)_t^i$ t: the tth proposal	$I(s)_1^{price}, I(s)_1^{leadtime}, I(s)_1^{quality}$ The 1st concession
	Buyer: $I(b)_t^i$ t: the tth counter-proposal	$I(b)_1^{price}, I(b)_1^{leadtime}, I(b)_1^{quality}$ t=1
Proposal	$P(s)_t = \langle I(s)_t^i \rangle$	$P(s)_1 = \langle I(s)_1^{price}, I(s)_1^{leadtime}, I(s)_1^{quality} \rangle$
Counter-proposal	$CP(b)_t = \langle I(\mathrm{b})_t^i \rangle$	$CP(b)_1 = \langle I(b)_1^{price}, I(b)_1^{leadtime}, I(b)_1^{quality} \rangle$
Negotiation sequence	$\{P(s)_t, CP(b)_t\}$	$\{P(s)_1, CP(b)_1, P(s)_2, CP(b)_2, P(s)_3, CP(b)_3, ...\}$

A convenient method to evaluate proposals and counter-proposals is to give a numerical score for each proposal and counter-proposal in the [0, 1] range. The numerical score can be perceived as the accumulated sum of individual scores for each negotiation issue. Among the negotiation issues, some quantitative issues are straightforward to obtain numerical scores. For instance, the value of lead time can be calculated as the amount of days. When the upper and lower value bounds are set, issue values can be normalized through the linear scoring function as in the second column in table 2. The expressions of scoring functions for benefit type issues (larger values are preferred) and cost type issues (smaller values are preferred) are differentiated. On the other hand, some qualitative issues are descriptive and not straightforward to be computed. Take the product quality as an example, different kinds of products may have different quality descriptions. For instances, textiles with higher density or higher ratio of natural fibre are considered to be of higher quality; while cast iron with lower deficiency is considered to be of higher quality. In this case, the issue values should belong to a finite set of descriptive options (e.g., when car's color is a negotiation issue, its value set may include black, red and white). Then, the fuzzy method with triangular fuzzy set is employed to calculate numerical issue values and scores. Similar fuzzy methods have also been used in [9, 15]. The third column in table 2 gives the specification of the fuzzy method and scoring function for qualitative issues.

Table 2. Specification of issue scoring functions

	Quantitative issue	Qualitative issue
A finite set of descriptive options	\	$Q = \{q_1, q_2, ..., q_k, ..., q_m\}$
Scoring function for descriptive options	\	$\overline{V} : Q \to [0,1]$
A fuzzy set for q_k	\	A triangular number (a_k, b_k, c_k)
Numerical issue value	$I_t^i \in [\min^i, \max^i]$	$I_t^i \in [\min\{a_1, a_2, \cdots, a_m\}, \max\{c_1, c_2, \cdots c_m\}]$
Fuzzy membership function	\	$\mu_{q_k}(I_t^i) = \begin{cases} (I_t^i - a_k)/(b_k - a_k) & a_k \le I_t^i < b_k \\ (c_k - I_t^i)/(c_k - b_k) & b_k \le I_t^i < c_k \\ 0 & otherwise \end{cases}$
Scoring function	$V(I_t^i) = \begin{cases} V_{\min}^i + (1 - V_{\min}^i) \times \left(\dfrac{I_t^i - \min^i}{\max^i - \min^i} \right) & \text{Benefit} \\ V_{\min}^i + (1 - V_{\min}^i) \times \left(\dfrac{\max^i - I_t^i}{\max^i - \min^i} \right) & \text{Cost} \end{cases}$	$V(I_t^i) = \sum_{q_k \in Q} \mu_{q_k}(I_t^i) \overline{V}(q_k)$

To calculate the numerical score of a proposal or counter-proposal, each composition negotiation issue should be assigned a weight (w^i). When a buyer evaluates a seller's proposal, the numerical score can be expressed as equation (1) shows. A seller's proposal is acceptable for the buyer if the score is no less than that of buyer's counter-proposal.

$$U\left(P(s)_t\right)=\sum_i w(b)^i \times V\left(I_t^i\right) \qquad (\sum_i w(b)^i=1) \tag{1}$$

2.2. Negotiation strategies and concession tactics

Agents' negotiation behaviors are governed by negotiation strategies. Game-theoretic based negotiation strategies have been explored to determine the optimal solution by analyzing the interaction as a game between identical participants and seeking its equilibrium [16, 17]. Most of these approaches assume that the space of negotiation issue values is completely known, however, this is not realistic in E-commerce negotiations where participants are reluctant to reveal all their compositions of negotiation issue values to their opponents. On the other hand, the space of negotiation issue values is infinite when consecutive issue values are involved (such as price), which will go beyond agents' limited computation capabilities to determine the optimal solution. Therefore, the heuristic computing approach suggested by Faratin, Sierra and Jennings [1] is adopted to formulate the negotiation strategies through continuous concessions. The combination of concession tactics involving time-dependent, resource-dependent and behavior-dependent tactics introduced in [1] is an applicable computational approach for the composition of negotiation strategies reflecting agents' attitudes towards risks, time limits and resource availability.

For the time-dependent concession tactic, time is the predominant factor used to decide how to make the concession. This tactic is based on continuous functions containing time variables. In the t^{th} (t=1, 2, 3…) proposal or counter-proposal, the concession rate for issue I^i can be defined as:

$$\alpha_i(t)=\begin{cases} k_i+(1-k_i)((t-1)/t_{\max})^{1/\beta} & (0<\beta\le 1)\\ \exp\left(\left(1-(t-1)/t_{\max}\right)^{\beta}\ln k_i\right) & (\beta>1)\end{cases} \tag{2}$$

Based on this concession rate, the value of issue I^i in the t^{th} proposal or counter-proposal can be expressed as:

$$I_t^i=\begin{cases} \min^i+\alpha_i(t)(\max^i-\min^i) & \textit{Cost}\\ \max^i-\alpha_i(t)(\max^i-\min^i) & \textit{Benefit}\end{cases} \tag{3}$$

In equation (2), $t_{\max}$ is the maximum willing concession step. $\max^i$ and $\min^i$ are the upper and lower value bounds of issue I^i. They can also be perceived as the reservation and aspiration issue values which are the least and best acceptable values respectively. In the commerce world, sellers usually make a higher discount on the initial proposal to smooth the following confronting, and then concede slowly. In the concession function, k_i reflects the discount rate. The variable β influences the concession speed. When 0<β<1, the concession speed is low in the initial stage, and then increases gradually. When β=1, the concession speed is a constant. When β>1, the concession speed is high in the initial stage, and then decreases gradually to a small value.

The resource-dependent concession tactics generate issue values depending on how a particular resource is being consumed. As time can be seen as a kind of resource, the resource-dependent concession function can be illustrated in the same way as time-dependent concession tactics.

The behavior-dependent concession tactics are also called imitative tactics. There are different approaches to imitate the opponent's behavior, namely, relative tit-for-tat tactic, averaged tit-for-tat tactic and random absolute tit-for-tat tactic [1]. The relative tit-for-tat tactic controls the imitation starting point and imitates the opponent's concession behavior according to the relative change of the opponent's previous two proposed issue values.

A negotiation strategy is a combination of the concession tactics either in the form of direct weighted sum, or periodic change of different tactics. A feasible negotiation strategy should be able to achieve the negotiation agreement within the allowable negotiation time and resources.

3. Adaptive Negotiation Behavior Configuration

In agent-based E-commerce negotiations, software agents carry out negotiations on behalf of either buyers or sellers. Agents' negotiation behaviors are configured firstly through the assignment of suitable negotiation strategies, and then may be adjusted by detecting opponents' behaviors during the negotiation. After the negotiation, the negotiation sequence has to be recorded. A three-staged adaptive behavior configuration mechanism is illustrated in figure 1. The mechanism is described from the buyer agent perspective, and is also applicable for the seller agent.

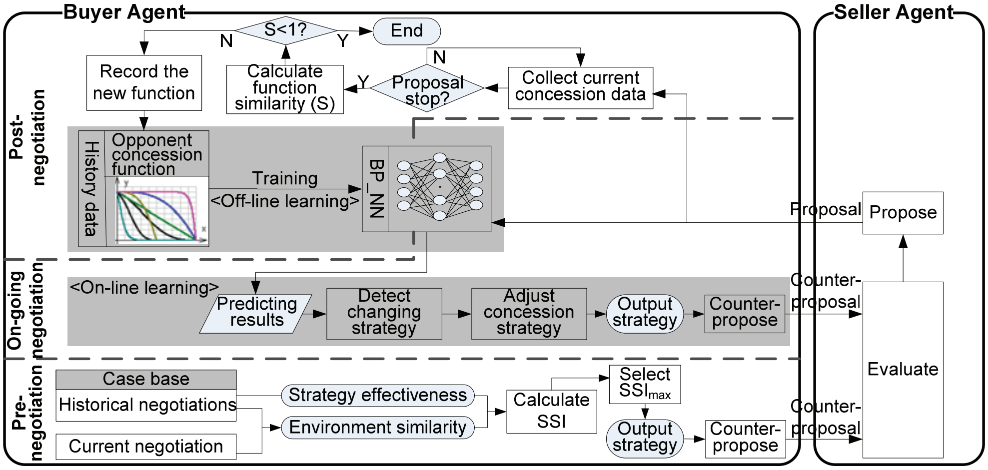


Fig. 1. Adaptive negotiation behavior configuration framework

3.1. Case-based pre-negotiation strategy assignment

In the negotiation behavior configuration framework, negotiation strategies are assigned learning from previous negotiation experiences. When eliciting adoptable strategies from the negotiation case base, the negotiation environment similarity and negotiation strategy effectiveness are considered as two measures. Some environmental attributes are chosen to specify the negotiation environment, such as negotiation issue, issue value range, total number of negotiation participants and negotiation time limit. The negotiation strategy effectiveness can be measured by the individual utility gained from a negotiation (which is also the numerical score of the final proposal or counter-proposal). Then, a strategy selection indicator (SSI) can be calculated combining the above two measures. The lower section in figure 1 shows the strategy deployment procedure in the following 5 steps.

Step 1: to determine the environmental attributes and historical individual utility.

These data are recorded in the historical negotiation case base and can be directly retrieved.

Step 2: to calculate individual environmental attribute similarities between the current and historical negotiations.

The environmental attributes may involve quantitative and qualitative attributes. They can be expressed as:

$x_{qni} \in [\min_{qni}, \max_{qni}]$: the range value of quantitative environmental attributes;

$x_{qli} \in \{FV_{qli1}, FV_{qli2}, \cdots, FV_{qlin}\}$: a set of fuzzy values of qualitative environmental attributes.

For the environmental attribute i, the individual similarity between the current attribute value and the attribute value of the jth historical negotiation is defined as:

$$\text{For quantitative attributes: } Sim_i^{C,H_j} = 1 - \frac{|x_i^C - x_i^{H_j}|}{\max\{x_i^C, x_i^{H_j}\}} \tag{4}$$

$$\text{For qualitative attributes: } Sim_i^{C,H_j} = \begin{cases} 1 & FV_i^{H_j} \le FV_i^C \\ 1 - [FV_i^{H_j} - FV_i^C] & otherwise \end{cases} \tag{5}$$

Step 3: to calculate the aggregated environment similarity.

The negotiation environment similarity can be calculated as a weighted sum of all the individual similarities:

$$\overline{Sim}^{C,H_j} = \sum_i w_i \times Sim_i^{C,H_j} \tag{6}$$

Step 4: to calculate the strategy selection indicator.

The strategy selection indicator for the jth historical negotiation can be calculated as the weighted sum of individual utility (iu) and the aggregated environment similarity:

$$SSI_j = w_{iu} \times iu_j + w_{sim} \times \overline{Sim}^{C,H_j} \quad (7)$$

Step 5: to select the maximum SSI value.

After calculating the SSI values of all the historical negotiations, the one with the maximum SSI value can be determined. The negotiation strategy used in this negotiation will be assigned to the agent as the current strategy.

3.2. Neural network-based negotiation behavior tracking

The Back-Propagation Neural Network (BP_NN) learning model is used to track the negotiation opponent's behavior. The fundamental logic is learning the opponent's historical concession behaviors in the off-line learning module to establish the BP_NN structure, and then in the on-line learning module, using the BP_NN to detect the dynamic change of opponent's behavior. The BP_NN can be trained using concession data series. Since the wide coverage of the data value scales may cause pattern confusion, the input data should be processed to eliminate that influence. Here, the input data are processed to be the ratio of difference between the successive raw concession data. The definition of the data is as follows.

$I(s)_t^i$: the value of negotiation issue I^i proposed by the seller at his concession step t, t =1, 2, 3….

$D_t^i = I(s)_t^i - I(s)_{t+1}^i$: the difference between two adjacent proposed values of issue I^i, also the concession value.

$R_t^i = D_t^i / D_{t+1}^i = \{I(s)_t^i - I(s)_{t+1}^i\} / \{I(s)_{t+1}^i - I(s)_{t+2}^i\}$: The ratio of difference.

Three concession trends can be learned in the BP_NN covering general attitudes towards risks, namely, risk prone (P), risk averse (A) and risk neutral (N). For the risk prone type, the concession speed is in a decreasing manner, agents concede aggressively at the start in order to reach the agreement as early as possible. For the risk averse type, agents keep a slow concession speed at the start to avoid the rapid benefit losing, and the concession speed is in an increasing manner. For the risk neutral type, agents always concede at a constant concession speed.

The parameter settings of the BP_NN are described in table 3. The network structure is of 4 input nodes, 12 hidden nodes and 4 output nodes. The input node x_in_0 represents the concession stage and is counted from 0.01. The input nodes x_in_1, x_in_2, x_in_3 are three successive ratios of difference. When there is a new ratio of difference input, the concession stage will move forward by an increment of 0.01. The output node y_out_0 is the predicted following ratio of difference. The output nodes y_out_1, y_out_2 and y_out_3 perform as a concession type classifier. (1, 0, 0), (1, 1, 0) and (0, 0, 1) stand for the risk averse, risk prone and risk neutral concession type respectively.

Table 3. BP_NN parameter settings

Parameter	Setting	
Network structure	4-12-4	
x_in_0	concession stage	
x_in_1, x_in_2, x_in_3	R_t, R_{t+1}, R_{t+2}	
y_out_0	R_{t+3}	
y_out_1	concession type classifier	(1,0,0)→A
y_out_2	concession type classifier	(1,1,0)→P
y_out_3	concession type classifier	(0,0,1)→N
Activation function in input layer	f(x)=x	
Activation function in hidden and output layers	$f(x) = 1/(1+e^{-x})$	
Performance measure (m training samples, q output nodes)	$RMSE = \sqrt{\sum_{k=0}^{m-1}\sum_{o=0}^{q-1}(d\_out_o(k) - y\_out_o(k))^2 / mq}$ (d_out_o : desired output)	

In the on-line learning phase, the BP_NN structure trained in off-line learning is used to predict the opponent's concession behavior and detect the changes of its concession functions. The behavior tracking logic is presented in figure 2. When confronting with a negotiation opponent in a new negotiation sequence, the opponent's proposed issue values are input into the BP_NN and then processed. For multi-issue negotiations, values of each issue are input into a separated BP_NN, and the concession functions of each issue can be detected in parallel. Since the output is still a ratio of difference, it should be converted back to the predicting issue value.

The relative difference between the predicted issue value and the actual issue value is computed to determine if there is fluctuation of the opponent's concession behavior. If some fluctuation has been detected, the opponent may probably change the concession function, and a new round of detecting and predicting will be conducted to capture the dynamic changes. For the three output nodes of the concession type classifier, if at least two nodes are stable (being close to 0 or 1 in a continuous manner), the concession trend can be identified to be similar to the concession functions in the negotiation case base. Otherwise, an unfamiliar new concession function may be encountered.

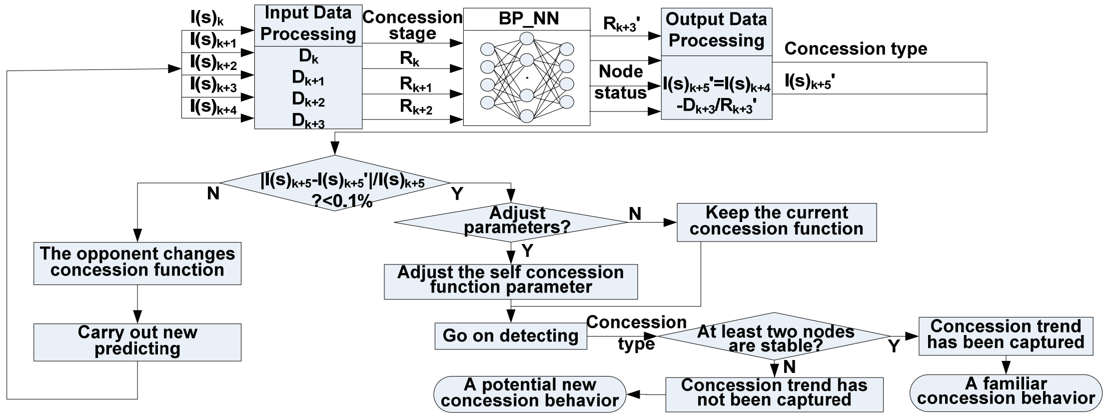


Fig. 2. Negotiation behavior tracking logic

3.3. Negotiation behavior recording

When a negotiation ends, all the opponent's proposed issue values will be summarized as a new concession time series. The difference between the shapes of the new time series curve (C) and a historical concession curve (H_j) can be calculated as a similarity measure. Supposing the total concession steps taken in a historical negotiation and new negotiation are T_H and T_C respectively. T_H and T_C can be divided into N time segments with each time segment involving T_H/N and T_C/N steps of concession respectively. In each time segment, the concession curve can be approximated to a straight line shaped by the starting and ending issue values. The slope of the straight line in the i^{th} time segment can be expressed as equation (8) for C and H_j respectively (symbol I represents the issue value):

$$k_i^C = \frac{I^C_{\frac{T_C}{N}\cdot i} - I^C_{\frac{T_C}{N}\cdot(i-1)}}{T_C / N} \text{ and } k_i^{H_j} = \frac{I^{H_j}_{\frac{T_H}{N}\cdot i} - I^{H_j}_{\frac{T_H}{N}\cdot(i-1)}}{T_H / N} \tag{8}$$

Then the similarity measure (S) of C and H_j is the mean square deviation of k_i^C and k_i^H as equation (9) shows.

$$S_C^{H_j} = \sqrt{\frac{\sum_{i=1}^{N}(k_i^C - k_i^{H_j})^2}{N}} \tag{9}$$

For the similarity measure between C and each H_j, if none of the $S_C^{H_j}$ is smaller than 1, it shows that the new concession time series is of apparent difference comparing with all the historical concession functions. Therefore, the new concession time series should be recorded, and the BP_NN needs to be trained again for updating.

4. Testing and Experiments

4.1. Training and testing of the BP_NN

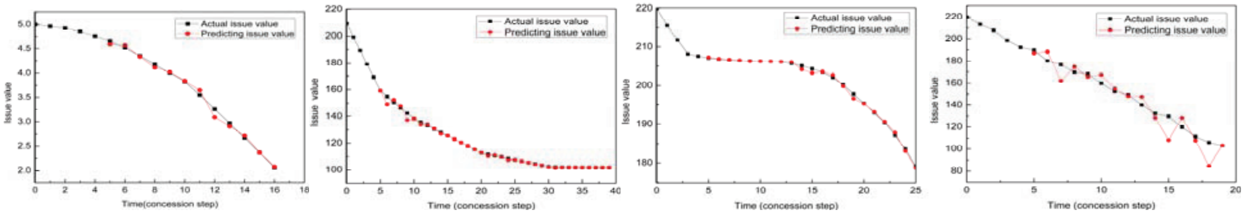
There are 440 initial training samples for the BP_NN. They are selected based on the time-dependent concession functions introduced in section 2.2 covering seven functions with various values of $t_{\max} \in \{50,100\}$, $\beta \in \{1/20, 1/3, 1, 3, 20\}$, $k_i = 0.1$, $\max^i = 300$ and $\min^i = 100$. Although the training samples are limited, they can reflect the typical concession trends of risk neutral concession tactic with constant concession speed (β=1), risk averse concession tactics with increased concession speeds (0<β<1) and risk prone concession tactics with decreased concession speeds (β>1). After 657,548 epochs of training, the training error (RMSE) is less than 0.01 and the network weights and threshold bias are determined.

To test the adaptability of the learning model, the testing data include time-dependent concession functions with different parameter settings and random concession patterns with and without regular trends. For the time-dependent concession testing data, three functions (a-c) with diversities of parameter settings are randomly chosen (as in table 4). For the random concession testing data, random concession values are generated by the Java random method in each concession step. These values may be generated in a totally random manner without any regular trends, or they can be relative random numbers following a general increased or decreased speed trend.

Table 4. Parameter settings of the testing time-dependent concession functions

	Concession type	t_{max}	β	k_i	max^i	min^i
a	risk neutral	20	1	0.05	5	2
b	risk averse	12	0.5	0.16	150	135
c	risk prone	80	6	0.28	2000	1500

For the time-dependent concession testing, the deviations between the predicting values and the actual values are quite small. For the random concession testing, figure 3 displays four examples of the concession curves and the related predicting curves from the BP_NN. For graphs 3(a)-3(c), the random concession values are controlled to go along the increase, decrease or a hybrid speed trend. For graph 3(d), the concession values are totally random values between 0 and 10. Table 5 shows the first 10 actual concession values of 3(b) and 3(d) as an example. For random concessions with trends (3(a)-3(c)), the general trends can be tracked while some predicting values are of obvious deviations from actual values, that is because the random values are not smooth enough to form the trends as strict concession functions. For random concessions without regular trends (3(d)), the predicting values are in disorder.



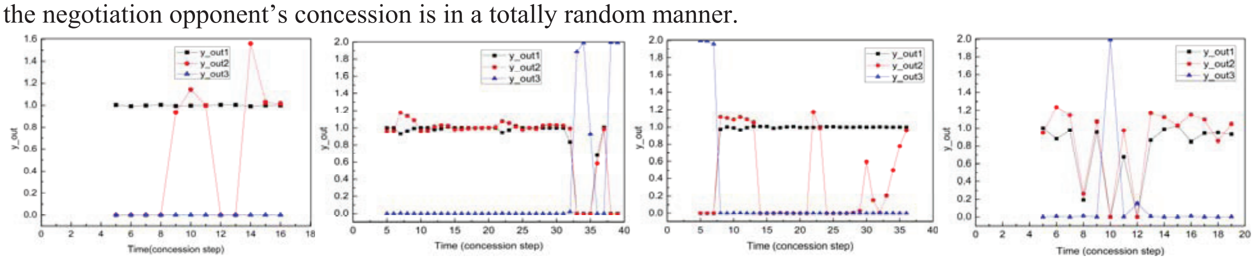
(a) increase (b) decrease (c) decrease-increase (d) random

Fig. 3. The comparison of predicting issue values and actual issue values for random concessions

Table 5. The examples of random concession values

	1	2	3	4	5	6	7	8	9	10
b	10.10785	10.0608	10.03223	10.03174	10.01133	4.345534	4.234194	4.118923	4.0902	4.081664
d	6.711582	5.368028	9.315115	6.353309	2.489479	9.546816	3.714945	6.597773	1.574748	8.457714

Figure 4 shows the concession type predicting results for the four random concession examples. Although the predicting results are not stable along the whole concession sequence, there are still subtle clues to identify the random concession trends. For the random concessions with trends, the relatively stable phases of the predicting results can figure out the periodic increase or decrease trend. While for the random concessions without trends, it is much difficult to identify the successive stable phase. If this phenomenon of chaos happens, it can be inferred that the negotiation opponent's concession is in a totally random manner.



(a) increase (b) decrease (c) decrease-increase (d) random

Fig. 4. The plotting of predicting concession types for random concessions

From the testing results of both the time-dependent concession functions and random concession patterns, it can be seen that the trends of the opponent's concession behavior can be captured by the BP_NN model regardless of a strict concession function or a random concession pattern being used.

4.2. An agent computational negotiation example

A two-issue buyer-seller negotiation case is implemented according to the computational method using JADE (Java agent development environment). Considering the multi-issue negotiation scenario, both the quantitative negotiation issue (unit price: up) and qualitative issue (quality level: ql) are involved in the experimental case. Firstly, the qualitative issue needs to be quantified for computing. Table 6 shows the quantification method in accordance with the introduction in section 2.1. The parameter settings of the experiment are described in table 7. In this case, both the buyer and seller agents use time-dependent concession functions, the seller agent will change its concession function parameters at its concession step t=20. The buyer agent can detect the changes of the seller's concession function and adjust its concession function parameters accordingly.

Table 6. Quantification method for the qualitative issue (quality level)

	Buyer	Seller
Descriptive options	{Level1, Level2, Level3}	{Level1, Level2, Level3}
Scores for descriptive options	{1, 0.7, 0.2}	{0.3, 0.6, 1}
Fuzzy sets for descriptive options	{(1,1,2), (1.5,2.5,3.5), (3,4,4)}	{(2,2,3), (2.5,3.5,4.5), (4,5,5)}
Issue value range	[1, 4]	[2, 5]
Fuzzy membership functions	$\mu_{Level1}(I_t^{ql})=\begin{cases}2-I_t^{ql} & 1\le I_t^{ql}<2\\ 0 & \text{otherwise}\end{cases}$	$\mu_{Level1}(I_t^{ql})=\begin{cases}3-I_t^{ql} & 2\le I_t^{ql}<3\\ 0 & \text{otherwise}\end{cases}$
	$\mu_{Level2}(I_t^{ql})=\begin{cases}I_t^{ql}-1.5 & 1.5\le I_t^{ql}<2.5\\ 3.5-I_t^{ql} & 2.5\le I_t^{ql}<3.5\\ 0 & \text{otherwise}\end{cases}$	$\mu_{Level2}(I_t^{ql})=\begin{cases}I_t^{ql}-2.5 & 2.5\le I_t^{ql}<3.5\\ 4.5-I_t^{ql} & 3.5\le I_t^{ql}<4.5\\ 0 & \text{otherwise}\end{cases}$
	$\mu_{Level3}(I_t^{ql})=\begin{cases}I_t^{ql}-3 & 3\le I_t^{ql}<4\\ 0 & \text{otherwise}\end{cases}$	$\mu_{Level3}(I_t^{ql})=\begin{cases}I_t^{ql}-4 & 4\le I_t^{ql}<5\\ 0 & \text{otherwise}\end{cases}$

Table 7. Experimental parameter settings

	Buyer		Seller	
Issue	Unit price	Quality level	Unit price	Quality level
Weight	0.8	0.2	0.7	0.3
Issue range	[150,300]	[1,4]	[180,320]	[2,5]
t_{max}	100	100	100	100
Starting concession function	β=0.5, k_{up}=0.1	β=2, k_{ql}=0.001	β=1, k_{up}=0.15	β=1, k_{ql}=0
Changed concession function	β=1/3, k_{up}=0.1	β=1, k_{ql}=0.001	β=1/3, k_{up}=0.15	β=0.5, k_{ql}=0
Changing point	t=32	t=32	t=20	t=20
Utility scoring function	$(300-I_t^{up})/(300-150)\times 0.8+V(I_t^{ql})\times 0.2$		$(I_t^{up}-180)/(320-180)\times 0.7+V(I_t^{ql})\times 0.3$	

The final negotiation results of two comparison experiments are shown in table 8. The utilities are compared between negotiations with and without buyer's dynamic detecting and adjustment of concession tactics. It can be observed from the outcomes that the buyer win higher utility if it tracks the opponent's concession behavior and adjust its own concession functions accordingly. At this point, the functioning of the computational negotiation is clarified. The negotiation power of the buyer agent can actually be enhanced through the tracking and adaptive adjustment of negotiation behaviors.

Table 8. The comparison of negotiation results

	Mutual accepted proposal	Buyer utility	Seller utility	Concession steps taken
Without dynamic detection	$CP(b)_{75}=\langle 240.94, 2.948\rangle \Rightarrow \langle 240.94, \text{Level2}\rangle$	0.3923	0.3900	75
With dynamic detection	$CP(b)_{78}=\langle 230.47, 2.979\rangle \Rightarrow \langle 230.47, \text{Level2}\rangle$	0.4438	0.4349	78

5. Conclusions

Agent technologies are being explored to automate E-commerce negotiations. As computational entities, agents carry out negotiations based on some computation model. This paper provides a computational method to organize agent-based E-commerce negotiations with adaptive negotiation behaviors. Negotiation issues and strategies are expressed in a computational pattern. Agents' negotiation behaviors are configured and tracked through a three-staged mechanism involving the case-based pre-negotiation strategy assignment, the neural network-based negotiation behavior tracking and the time series-based post-negotiation data recording. Through this mechanism, agents' negotiation behaviors can be deployed in a more adaptive and flexible manner. The computational negotiation method has been implemented using an assumed two-issue buyer-seller negotiation case. Observing from the experimental results, the buyer can actually win more benefit if the opponent's negotiation behavior is tracked at runtime. At this stage, the computational model configures the essential negotiation functionality of an individual agent, for the future, the allocation of the agent system in both business and customer ends will be specified with more real-life application considerations. Meanwhile, more negotiation concession patterns and dynamics will be explored to further test the adaptability of the negotiation behavior tracking and recording mechanisms.

References

1. P. Faratin, C. Sierra and N. R. Jennings, (1998). Negotiation decision functions for autonomous agents. International Journal of Robotics and Autonomous Systems, 24(3-4), 159-182.

2. V. Narayanan and N. R. Jennings, (2005). An adaptive bilateral negotiation model for e-commerce settings. In: 7th IEEE International Conference on E-Commerce Technology, Munich, Germany.

3. C. C. Huang, W. Y. Liang, Y. H. Lai and Y. C. Lin, (2010). The agent based negotiation process for B2C e-commerce. Expert Systems with Applications, 37(1), 348-359.

4. G. Weiss, (1999). Multiagent systems: a modern approach to distributed artificial intelligence, Cambridge, Mass.: MIT Press.

5. A. Chavez and P. Maes, (1996). Kasbah: An agent marketplace for buying and selling goods. In: First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London.

6. P.R. Wurman, M.P. Wellman and W.E. Walsh, (1998). The Michigan Internet AuctionBot: a configurable auction server for human and software agents. In: Second International Conference on Autonomous Agents, Minneapolis.

7. J. Collins, W. Ketter and M. Gini, (2002). A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. International Journal of Electronic Commerce, 7 (1), 35-57.

8. M. Louta, I. Roussaki and L. Pechlivanos, (2008). An intelligent agent negotiation strategy in the electronic marketplace environment. European Journal of Operational Research, 187(3), 1327-1345.

9. Y. M. Chen and P. N. Huang, (2009). Agent-based bilateral multi-issue negotiation scheme for e-market transactions. Applied Soft Computing, 9(3), 1057-1067.

10. R.Y.K. Lau, (2007). Towards a web services and intelligent agents-based negotiation system for B2B eCommerce. Electronic Commerce Research and Applications, 6(3), 260-273.

11. M. Oprea, (2002). An adaptive negotiation model for agent-based electronic commerce. Studies in Informatics and Control, 11(3), 271-279.

12. C. M. Hou. (2004). Predicting agents tactics in automated negotiation. In: 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04), Beijing.

13. J. G. Baek and C. O. Kim, (2007). Learning single-issue negotiation strategies using hierarchical clustering method. Expert Systems with Applications, 32(2), 606-615.

14. F. H. Ren, M. J. Zhang and K. M. Sim, (2009). Adaptive conceding strategies for automated trading agents in dynamic, open markets. Decision Support Systems, 46(3), 704-716.

15. L. Mikhailov, (2002). Fuzzy analytical approach to partnership selection in formation of virtual enterprises. The International Journal of Management Science, 30, 393-401.

16. S. Kraus, (2001). Strategic Negotiation in Multiagent Environments. Cambridge, Mass: MIT Press.

17. T. Sandholm, (2002). Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence, 135(1-2), 1-54.